Websocket SSL

By default, WCS uses the port 8080 to receive Websocket connections and the port 8443 to receive Secure Websocket connections.

To connect to the WCS server via the Secure Websockets protocol using Web SDK, pass the urlServer parameter as wss to the Flashphoner.createSession() function.

Example:

```
Flashphoner.createSession({urlServer:'wss://192.168.1.5:8443'});
```

Settings

Secure Websockets is managed by the followingsettings in the flashphoner.properties file:

Setting	Default value
wss.port	8443
wss.keystore.file	wss.jks
wss.keystore.password	password
wss.cert.password	password

Generated certificates (self-signed)

By default, WCS uses simple generated certificates, so in order to make connection successful you should preliminarily access this URL from the browser:

https://192.168.1.5:8443/

Insert the actual address of your WCS server in place of 192.168.1.5.

The browser will display a warning that the security certifixate the WCS server uses is not known to the browser or the OS. This is normal, because for the sake of testing a generated certificate is used. Depending on the browser used, either continue opening this address or add this address to the list of security exclusions. After that, the WCS client certificate will be cached by the browser and all subsequent connections will pass through successfully.

Valid certificates

Self-signed certificates can be used for testing and development only, because otherwise your users would have to perform a procedure of confirming the unknown certificate prior to connecting.

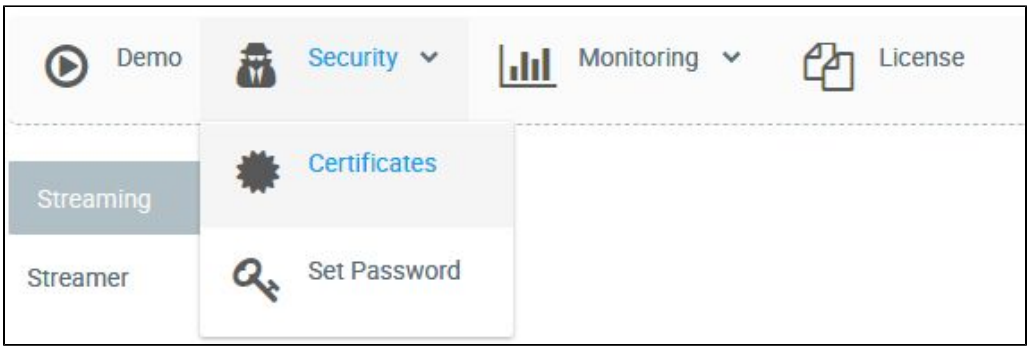
Production requires SSL certificates issued by an authorized certification center. Such certificates are usually paid and you can purchase them for a domain or a subdomain. Certificates must be imported to the key store wss.jks. See theSettingssection.

Import SSL certificates using web interface

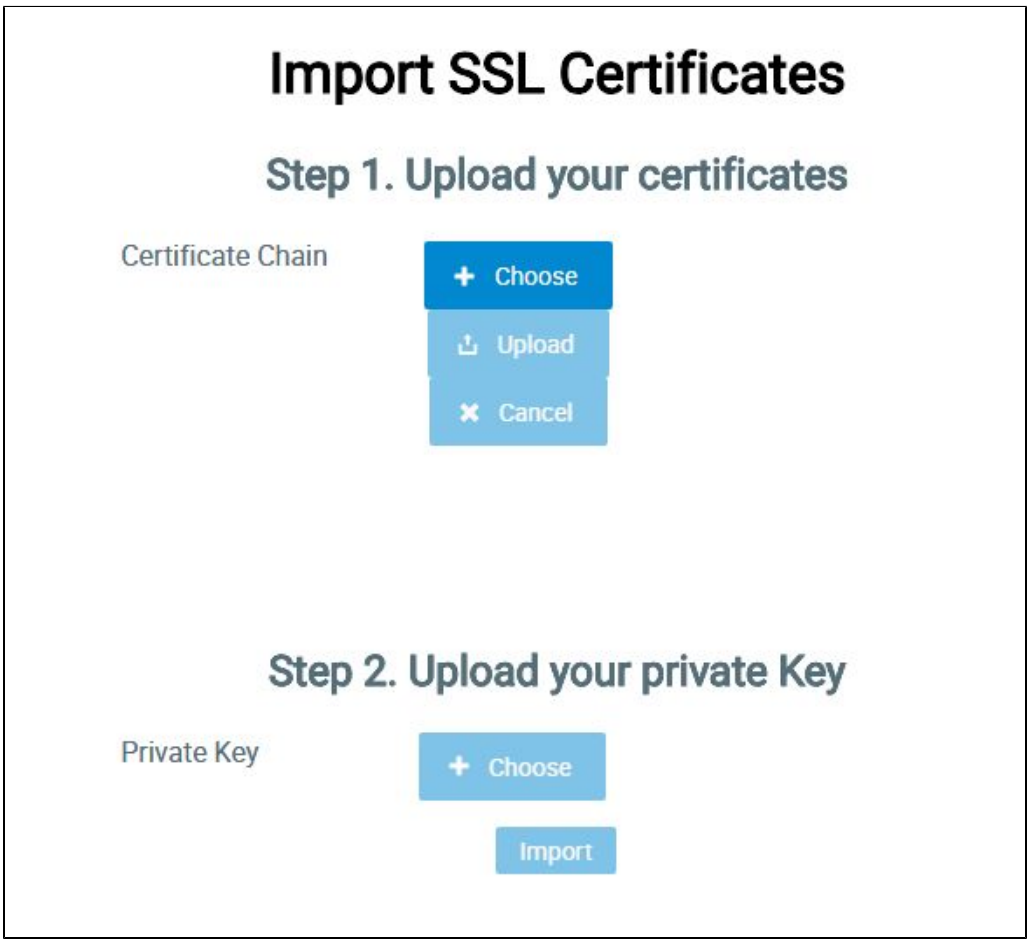
You can import an SSL certificate using the WCS web interface as follows:

1. Obtain an SSL certificate from your SSL provider.

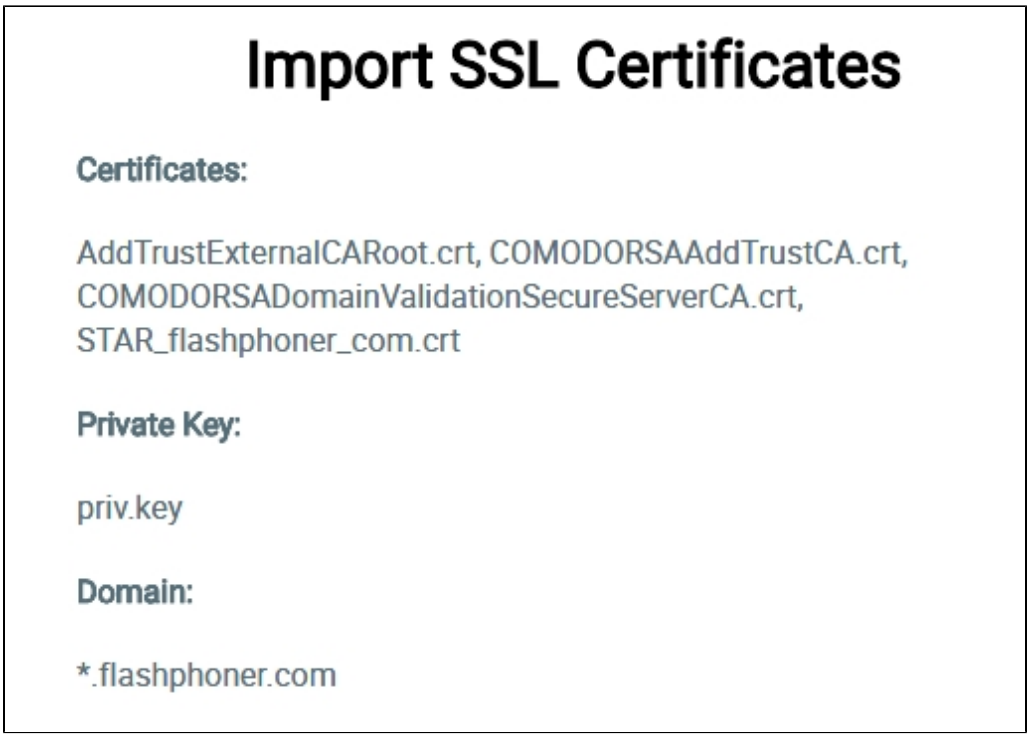
2. Open the web interface of WCS. Select "Security" in the upper menu, then the "Certificates" submenu:



3. On the import page upload certificate files received from your SSL provider and the private key file:



4. SSL certificates imported using the web interface are stored to the certificate storage wss.jks and to the database of the WCS server, and are then displayed in the "Security - Certificates" section:



Restart the WCS serverto apply new settings. After restart, openhttp://yourdomain:8443. If the certificate was imported correctly, you should see the browser accepts the WCS server certificate.

Import SSL certificate using keytool

If you cannot import an SSL certificate using the web interface, you canuse the keytool instead. Keytool command line utility is shipped with JDK. If you do not have JDK installed, please refer to theInstalling JDKsection. The keytool executable is found in the JDK_HOME/bin catalog, for example, /usr/java /default/bin. For more convenience you can create a link:

```
ln -sf /usr/java/default/bin/keytool /usr/bin/keytool
```

Also, you may needopensslto convert certificate files. If openssl is not installed, install it using this command

```
yum install -y openssl or apt-get install openssl
```

Obtain the SSL certificate from your SSL provider.

Below we describe importing a certificate from theStartSSLprovider.

From the provider we received a certificate for the test.flashphoner.com domain and the following set of files:

test.flashphoner.com.crt - file of the certificate
test.flashphoner.com.key - private key file
ca.pem - root certificate
sub.class2.server.ca.pem - intermediate certificate

Then, we perform the following 5 steps:

1. Remove the self-signed certificate from the keystore

```
keytool -delete -alias selfsigned -keystore /usr/local/FlashphonerWebCallServer/conf/wss.jks
```

2. Create a new keystore based on the certificate and the private key

```
openssl pkcs12 -export -out test.flashphoner.com.p12 -inkey test.flashphoner.com.key -in test.flashphoner.com.
crt -certfile ca.pem -certfile sub.class2.server.ca.pem -name test.flashphoner.com
```

At this stage you need to enter the password from your private keytest.flashphoner.com.key as well as set a password for the keystore itself. Here we set 'password'.

```
Enter pass phrase for test.flashphoner.com.key: ******
Enter Export Password: password
```

Notice that files of certificates may have different file extensions. Also, the certification center may provide one bundle certificate instead of individual root and intermediate certificates. In this case in the openssl command above you need just the -certfile option.

3. Import the newly created keystore to the existing wss.jks

```
keytool -importkeystore -srckeystore test.flashphoner.com.p12 -srcstoretype PKCS12 -destkeystore /usr/local
/FlashphonerWebCallServer/conf/wss.jks
```

At this stage you have to enter passwords from the imported keystore and the wss.jsk keystore.

```
Enter destination keystore password: password
Enter source keystore password: password
Entry for alias test.flashphoner.com successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled
```

Done. Certificates are successfully imported to the keystore.Restart the WCS serverto apply the new settings.

After restarting the server, openhttp://test.flashphoner.com:8443again. If certificates are imported correctly, you should see that the browser accepts the WCS server certificate.

Further you need to use the URL wss://test.flashphoner.com:8443 URL to connect to the server using the 'connect' method. Notice that we use the domain name here, not the IP address. The certificate issued for that name was imported to the keystore and is used by the WCS server. The test.flashphoner.com domain that we used in the example should be replaced by your own domain that the SSL certificate is issued to.

Import two or more certificates for different domains

Sometimes it is necessary to allow user access to the same server with different domains signed by different certificates. In that case all certificates should be imported withkeytool, and:

1. Access to server functions through SSL (Secure Websocket, HTTPS) will work for all of this domains.

2. Access to web interface through SSL will work only for one domain whose certificate was imported first.